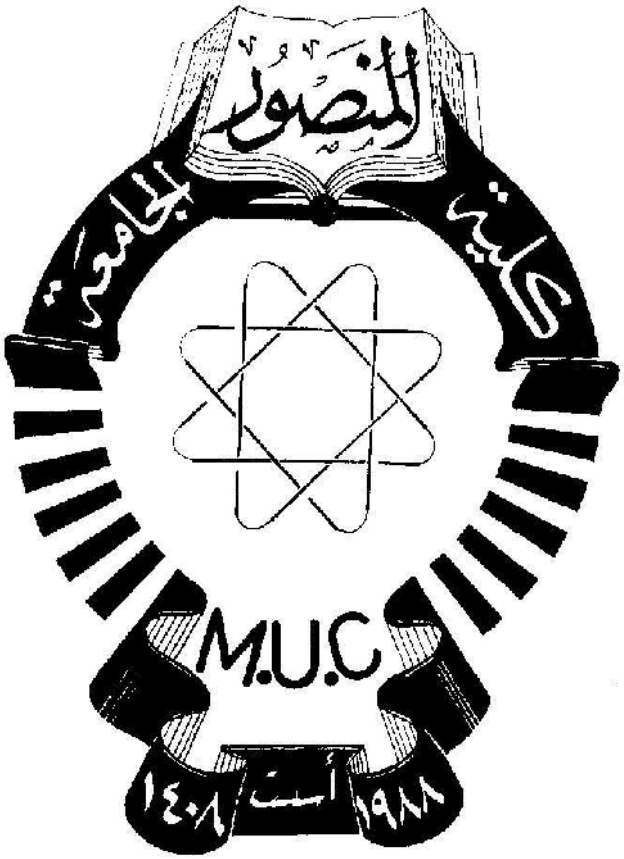
AL-MANSOUR JOURNAL

13th year –Issue(19)-2013

ISSN 1819-6489



19

A REFEREED SCIENTIFIC JOURNAL OF M.U.C Baghdad-Iraq
Member of International Association of Universities
Member of Federation of Arab Scientific Research Councils

Contents

Articles Published In Arabic

Articles Published In English

Development of Fast Reliable Secure File Transfer Protocol (FRS-FTP)

Dr. Mazin Sameer Al-Hakeem*, Suhiar M. Zeki*, Sarah Y. Yousif *

Abstract:

There is a great need to transfer information between hosts and networks in fast with reliable and secure way; this is a big challenge with open environment especially Internet and TCP based Networks. There are several extra file transfer protocols behind core FTP protocol, but each of them suffer from either slow, unreliable or unsecure workflow.

In this paper, we develop a new file transfer protocol based on UDP as a fast, reliable and secure protocol; and called FRS-FTP (refers to Fast Reliable Secure File Transfer Protocol. The proposed protocol based on three phases, the first phase to enforce the "reliability" issue using cryptographic hash checksum, the second phase to enforce the "security" issue using file encryption/decryption, port protection and authentication, while the third phase to transfer files with "fast" issue under UDP.

The proposed protocol is implemented using Visual Basic .Net programming language and System.Net.Sockets embedded dot Net Class.

Keywords: FRS-FTP, FTP, UDP, Secure Tunnel, Port Protection.

* Computer Science Department, University of Technology/ Baghdad

1. Introduction:

In computer world, there is a great need to transfer information between computer users through TCP-based network, such as the Internet [1].

There are many core application protocols that are used in various type of processing such as HTTP (Hyper Text Transfer Protocol) which is used to transfer hypertext and hypermedia over web information systems, SMTP (Simple Mail Transfer Protocol) which is used for e-mail transmission between devices, RTP (Realtime Transport Protocol) which is used for delivering audio and video over networks, and FTP (File Transfer Protocol) which is used to move data from one machine to another [1]. Some of these protocols (like HTTP, SMTP, FTP, etc.) are depending on TCP protocol which is reliable and connection-oriented, and others (like RTP, etc.) are depending on UDP protocol which is unreliable and connectionless-oriented. All of these core application protocols depend on the TCP/IP Reference model (which called also IP Stack) as shown in figure (1) [2].

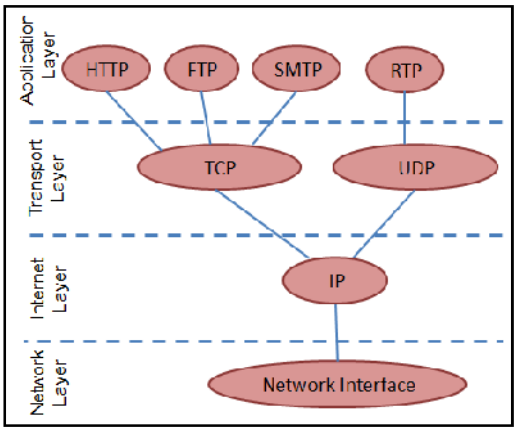


Figure (1): Some of Core Protocol Layering in TCP/IP Model.

FTP is commonly used to transfer files between hosts and networks, but FTP has some drawbacks such as slow and low security. The FTP is spent a long time for control transactions during transfer data (stop and waiting each block to be receive), so it's a slow protocol, but for the same reason its reliable protocol. The FTP login uses a week 'clear-text sign-in' authenticate protocol in the form of a plan username and password, so it has low security schema [2].

After sharp review for the network protocols of files transfer services (which describe in paragraph 4) and others academic researches, the closely related research to enhance FTP fast or security issues are addressing as the following:

In 2011, Taif Sami Hassan and others [3] develop AFTP (Adaptive File Transfer Protocol) which depends on UDP Protocol to transfer information, and use CRC (Cyclic Redundancy Check) to guarantee information receiving. MS Visual Basic and WinSock Control was used to implement the proposed AFTP. The proposed work does not suffer from stop and wait procedure drawback, and use CRC to ensure reliability (Fast and Reliable), but when an error occurs these means a need for more time to resend a message again (Slow in this case), and AFTP did not take into account the usage for any encryption function to secure the transmitted information between hosts and networks (Unsecure). AFTP is fast, in average case, and reliable but unsecure protocol.

In 2000, The Quest Software Company [4] develop SCP (Secure Copy Protocol) which depends on TCP Protocol to transfer the information, and use Secure Shell (SSH) protocol to enforce encryption functionalities over any transmitted information. Despite the SCP is enforce encryption policy to secure the transmit information between hosts and networks (Secure and Reliable), but it still suffers from stop and wait procedure drawback (Slow). SCP is secure and reliable but slow protocol.

2. TCP/IP Reference Model:

TCP/IP Reference model (also called IP Protocol Stack) is a set of communication protocols created in 1970s to enable computers to communicate over Internet and other similar networks. TCP/IP model provides end-to-end connectivity specifying how data should be formatted, addressed, transmitted, routed and received at the destination. This model is commonly known as TCP/IP, because of its most important protocols TCP and IP, which were the first networking protocols defined in this standard [5].

TCP/IP protocol has four layers and several core protocols, as describe below [1]:

- **Application Layer, it's a top TCP/IP layer, it's contains all the higher-level core protocols which involve user interaction, like HTTP, FTP, SMTP, RTP, DNS, BGP, RIP, and other core protocols.**
- **Transport Layer, it's located above the Internet Layer in the TCP/IP model, transport layer allow peer entities on the source and destination hosts to carryon a conversation, and provides the port number. It's used to exchange data between systems. This layer contains TCP and UDP Protocols.**
- **Internet Layer, it's located above the Transport Layer in the TCP/IP model, it's defines an official packet format and provides the IP Addressing (using core IP Protocol), to deliver after that independently through the network between the sender and receiver. The Internet Layer delivers "IP packets" (where they are supposed here) as is avoiding congestion.**
- **Network Interface Layer, it's interfaces the TCP/IP protocol stack to the physical network. It's a point out connect to the network using TCP/IP protocol, so it can send IP packets over it.**

3. TCP and UDP Transport Layer Protocols:

The transport layer allows exchanging data between hosts and systems over TCP-based network, such as the Internet. This layer

which called also Host-to-Host layer or End-to-End layer contains two main core protocols [1]:

- **TCP (Transmission Control Protocol): it is a reliable connection-oriented protocol, that offers error correction when deliver the stream originating on one machine to the other machine in the networks. So it used to send important data such as webpages, database information, etc;**
- **UDP (User Datagram Protocol): it is an unreliable connectionless protocol, that deliver stream originating on one machine to the other machine in the network based one-shot philosophy. So it used to send streaming audio and video (like Windows Media Audio files (.WMA), Real Player (.RM)).**

The developers were able to defines and add 37 extra protocols to work in transport layer, 4 of them for error detection and correction issue (like Alternant code, Casting out nines, Hardened Core), 10 for flow control (like Brooks Instrument, Ethernet flow control, Wormhole switching), 16 for network socket (like Berkeley Sockets, Socket API, WinSock), and so on [5].

4. The Network Protocols of Files Transfer Services:

4.1 FTP Application Layer Protocol:

The application layer contains all the higher-level protocols which involve user interaction, one of these protocols is FTP protocol which based on TCP/transport layer to provides a way to move data efficiently from one machine to another [5].
FTP is built on client/server architecture for upload files, webpages and other documents from a private development machines to a public files and web-hosting servers [5].
FTP uses separate control and data connections for transfer files between client and server depending on the following scenario and as shown in figure (2) [6]:

- **The FTP server will be running and waiting for incoming requests.**

- **The client computer is then able to communicate with the FTP server on port 21, this connection called the control connection, remains open for the duration of the session.**
- **A second connection, called the data connection, can either be opened by the server from its port 20 to a negotiated client port, or by the client from an arbitrary port to a negotiated server port as required to transfer file data.**
- **The control connection is used for session administration (for example commands, identification and passwords exchanged between the client and the server).**
- **The client computer uses port:1043 for control connection, and port:1045 for data connection.**
- **Due to this two-port structure, FTP is considered an out-of-band protocol, as opposed to an in-band protocol such as HTTP.**

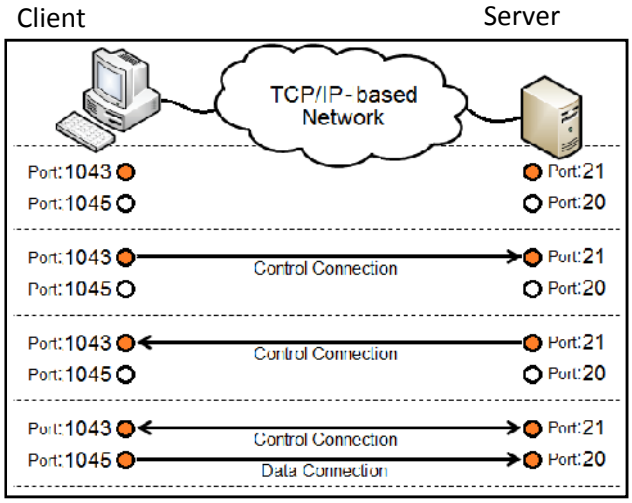


Figure (2): FTP Session Connections.

The first FTP client applications were interactive command-line tools, but today FTP client applications have GUI (Graphical User Interface), including general web design programs like Microsoft Expression Web, and specialist FTP clients such as CuteFTP. FTP login uses a 'clear-text sign-in' authenticate protocol, normally in the form of a username and password, for granting access but can connect

anonymously if the server is configured to allow it, and the session will commence [6].

4.2 Lesser-known Files Transfer Network Protocols:

The following list consists of the Lesser-known network FTP protocols:

1. **SFTP (Simple File Transfer Protocol) is a network protocol of files transfer services, which based on TCP, developed by Ward Christensen (founder of Computerized Bulletin Board System) in 1977 for personal use. SFTP is an unsecured file transfer protocol [7].**
2. **AFP (Apple Filing Protocol) is a network protocol of files transfer services, which based on TCP, developed by Apple Inc in 1988 for connecting their Mac operating system [8].**
3. **Lynx is a network protocol of files transfer services which based on UDP, developed by Matthew Thomas in 1989 for batch data transmission [9].**
4. **9P (Plan 9 Filesystem Protocol) is a network protocol of files transfer services which based on TCP, developed by Bell Labs in 1990 for connecting the components of distributed operating system on UNIX, it was developed primarily for research purposes [10].**
5. **BiModem is a network protocol of files transfer services which based on UDP, developed by Erik Labs in 1995 for use in Bulletin Board Systems. (Also called BBS, it is an online service based on microcomputers running appropriate software) [11].**
6. **SCP (Secure Copy Protocol) is a network protocol of files transfer services, which based on TCP, developed by ScriptLogic subsidiary of Quest Software Company in 2000 for commercial use. SCP is based on the Secure Shell (SSH) protocol to enforce encryption functionalities over any transmitted information [12].**

Some of these protocols have been developed to an advance copy in subsequent years and they are still in use.

5. The Proposed Fast Reliable Secure File Transfer Protocol (FRS-FTP):

5.1. The Proposed FRS-FTP Model:

Essentially, the proposed model for File Transfer Protocol (which called FRS-FTP) is based on three phases as shown in figure (3):

- **Phase I: To enforce the "reliability" issue using cryptographic hash checksum.**
- **Phase II: To enforce the "security" issue using file encryption/decryption, port protection and authentication.**
- **Phase III: To transfer files with "fast" issue under UDP.**

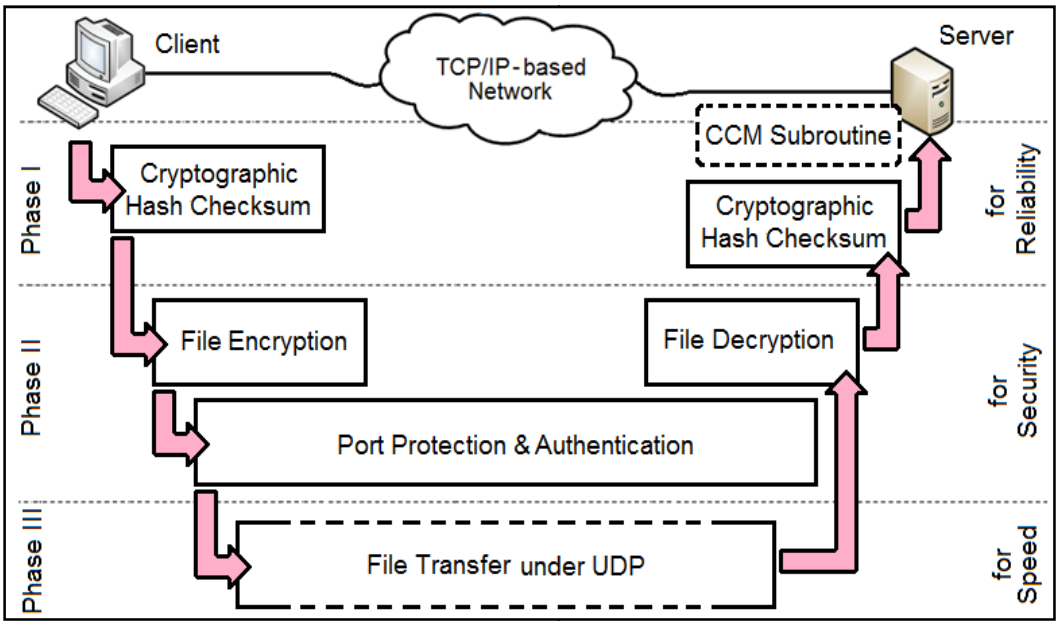


Figure (3): General Model for the Proposed FRS-FTP Protocol.

5.1.A. Phase I – Reliability Phase:

To enforce the "reliability" issue to check if the error occurred when transferring data, the checking will be done using cryptographic hash checksum function (sometimes called a message digest function) that takes an arbitrary block of data and returns a fixed-size bit string called checksum.

There are several methods to use a block cipher to build a cryptographic hash function, specifically a well-known one-way

compression function including MD4, MD5, SHA-1 and SHA-2. In this paper MD5 was used as a cryptographic hash checksum function, with the following specification as shown in table (1):

Table (1): MD5 Specifications

Specification	Value
Output size (bits):	128 bits
Internal state size:	128 bits
Block size:	512 bits
Length size:	64 bits
Word size:	32 bits
Collision and Preimage attacks (complexity):	Yes

The sender calculate the data checksums using MD5 as an offline process, and the receiver will recalculate the received data checksums then compare if equal or not. If error occurred, the error block must be resent directly without delay by waiting each block to be received (without stop and wait protocol).

5.1.B. Phase II – Security Phase:

Phase II will enforce the "security" issues as following to establish a "secure peer-to-peer tunnel".

1. **The proposed protocol uses DES encryption/decryption file algorithm (with ke sizes: 56-bits, block sizes: 64-bits, rounds: 16) as tool for providing privacy. The sender encrypts and receiver decrypts the data as an offline processes.**
2. **To prevent a serious vulnerability to dials-in port access and protect it, the proposed protocol uses "automatic call-back" process.**
   An authorized user dials a remote computer (receiver), after the user identifies him (using traditional password); the remote computer breaks the communication line. The remote computer then consults an internal table of calling numbers and calls the user back at a predetermind number.

This process will be achieved for port protection.

3. **The proposed protocol uses a public key cryptography to authenticate the remote computer and allow it to authenticate the users.**
   The public key is placed on all computers that must allow access to the owner of the matching private key. The private key is never transferred through the network during authentication process step.
   This process will be achieved for secure tunnel.

5.1.C. Phase III:

In this phase, the data will be dividing into 64-bit blocks, and sent directly through the established secure peer-to-peer tunnel under UDP without delay by waiting each block to be received (without stop and wait protocol), so it's fast.

5.2. Core Control Message (CCM) Subroutine:

The implemented Core Control Message (CCM) Subroutine is important core subroutine which works together with proposed FRS-FTP protocol. CCM Subroutine used by the remote computer (receiver) to send error messages indicating for the following statuses:

- **The requested FTP port is not available (ports 20, 21, 1043 and 1045).**
- **The computer could not be reached (connection disable).**
- **Recalculate the received data checksums is not equal to the received one (error occurred), as shown in figure (4).**

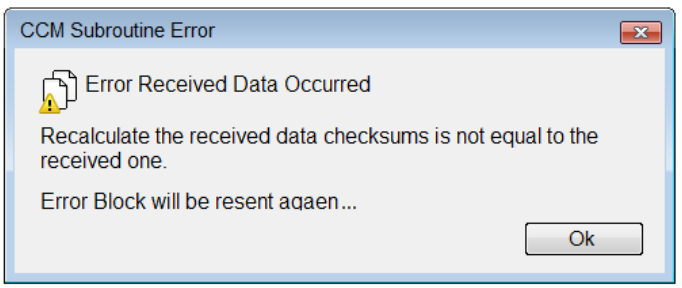


Figure (4): CCM Subroutine's Error Message.

5.3. Implementation of the Proposed FRS-FTP:

The proposed FRS-FTP protocol is implemented using Visual Basic .Net programming language and System.Net.Sockets embedded dot Net Class.

In this paper, the proposed FRS-FTP protocol works just with text files. The MD5 function was used to calculate the cryptographic hash checksum for text file and DES algorithm was used to encrypt and decrypt the text file. As a future work, we can expand the work to deals with multimedia files.

Figure (5) shows the screenshot of the implemented FRS-FTP System and figure (6) shows the CCM Subroutine message for successfully sent.

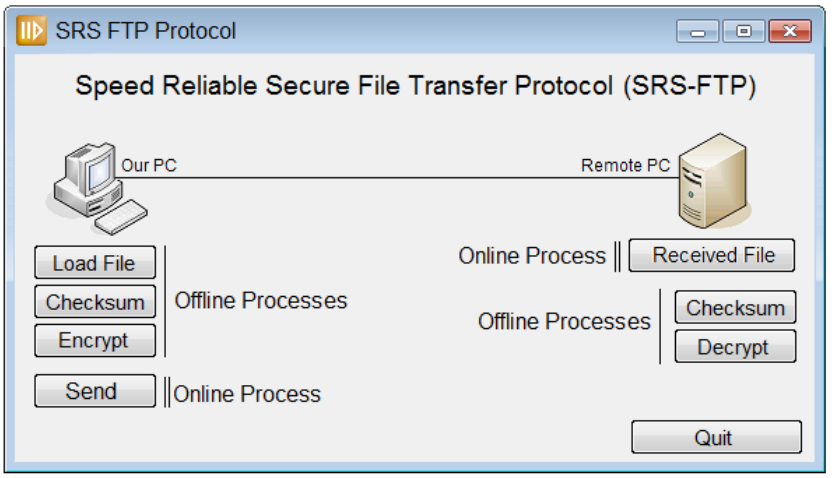


Figure (5): The Screenshot of FRS-FTP System.

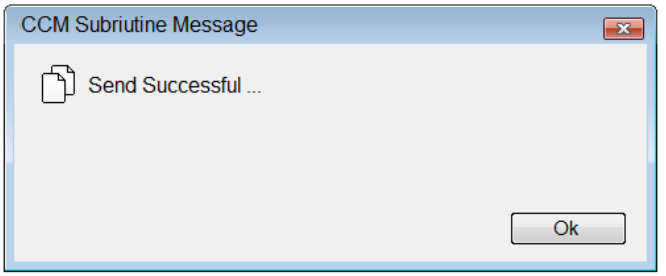


Figure (6): The CCM Subroutine Message for Successfully Sent.

5.4. Comparison between FRS-FTP and other Files Transfer Protocols:

Table (2) gives a comparison between network protocols of files transfer services, which presented in this paper, depending on "Fast", "Reliable" and "Secure" transfer's factors.

Table (2): Comparison of Files Transfer Protocols

Protocol Name	based on Protocol	Developer	Transfer's Comparison Factors		
			Fast	Reliable	Secure
FTP	TCP	American Army for ARPANET, 1971	No	Yes	No
SFTP	TCP	Ward Christensen, 1977	No	Yes	No
AFP	TCP	Apple Inc, 1988	No	Yes	No
Lynx	UDP	Matthew Thomas, 1989	Yes	No	No
9P	TCP	Bell Labs, 1990	No	Yes	No
BiModem	UDP	Erik Labs, 1995	Yes	No	No
SCP	TCP	Quest Software Company, 2000	No	Yes	Yes
AFTP	UDP	Taif S. Hassan and others, 2011	In average case Yes	Yes	No
The proposed FRS-FTP	**UDP**	**Authors, 2012**	In average case **Yes**	**Yes**	**Yes**

As shown in table (2), the proposed file transfer protocol (FRS-FTP) achieves fast, reliable and secure features over file transfer idea.

6. Conclusions:

The following points represent the important conclusions which are drawing through the development of the FRS-FTP Protocol:

1. **There is no protocol does work well from the first time, because it defined, implemented and reviewed by fallible humans. This one of the biggest protocol flaws.**
2. **The proposed protocol is effective more than other network protocols for files transfer services, and the proposed model (in figure (3)) illustrates how uses a combination of controls to achieved the proposed FRS-FTP Protocol with fast, reliable and secure facilities.**
3. **To the proposed protocol:-**
   - I. **The offline MD5 (cryptographic hash checksum) function provides the "reliability".**
   - II. **DES encryption/decryption file algorithm provides the "privacy" for transmitted data.**
   - III. **The automatic call-back process enforces the "port protection" and "prevents" a serious vulnerability to dials-in port access.**
   - IV. **The authentication process using public key cryptography enforces "secure peer-to-peer tunnel" between the sender and receiver.**
4. **The transfer files process under UDP provides a "fast way" to transfer the data between the sender and receiver.**

7. Recommendations:

Below are some recommendations for future work:

1. **Expanding the FRS-FTP Protocol to transfer multimedia files between hosts and networks.**
2. **Upgrading the FRS-FTP Protocol to capture and analyze the transferred data from one machine to another.**
3. **Developing the FRS-FTP Protocol to filter the attached data in order to Firewall and Anti-Virus scheme.**

8. References:

1. Tanebaum, A. S., "Computer Network", 3rd Edition, Prentice Hall International Inc, 2010.
2. Deepankar Medhi and Karthikeyan Ramasamy, "Network Routing: Algorithms, Protocols, And Architectures", Morgan Kaufmann, 2007.
3. Taif S. H., Alyaa H. A. and Alaa N. M., "Development of an Adaptive File Transfer Protocol", Engineering and Technology Journal, Vol.29, No11, 2011.
   Available at: www.uotechnology.edu.iq/tec_magaz/ volum292011/ No.11.2011/text/Text%20%2812%20%29.pdf.
4. Quest Software Company's Secure Protocol,
   Available at: www.quest.com/secure-copy/
5. Roman Dunaytsev and Dmitri Moltchanov, "TCP Performance Modeling in Wired and Wired/Wireless Networks", LAP Lambert Academic Publishing, 2011.
6. Website available at: www.slacksite.com/other/ftp.html.
7. Lambert M. Surhone, Mariam T. Tennoe, and Susan F. Henssonow, "Simple File Transfer Protocol", Betascript Publishing, 2010.
8. AppleShare and AppleShare IP File Sharing,
   Official Website available at: support.apple.com/kb/TA21611? viewlocale=en_US.
9. Official Website available at: www.cpeterso.com/code/protocols/ LYNX.TXT.
10. Official Website available at: 9p.cat-v.org.
11. Lambert M. Surhone, Mariam T. Tennoe and Susan F. Henssonow, "Punter (protocol)", Betascript Publishing, 2010.

Janp, "How the SCP Protocol Works", Jan Pechanec's weblog, 2007

تطوير اتفاقية نقل الملفات السريعة والموثوقة والأمنة

م.د.مازن سمير الحكيم ، م.م. سهير محمد زكي ، الباحثة سارة يعقوب يوسف

المستخلص

هناك حاجة كبيرة لنقل المعلومات بين الاجهزة المضيفية والشبكات بسرعة وبطريقة آمنة وموثوقة، وهذا يعد بمثابة تحد كبير خصوصاً عند العمل في بيئة مفتوحة مثل الإنترنت والشبكات المستندة على TCP . هناك العديد من أتفاقيات نقل الملفات بالاضافة الى اتفاقية FTP الأساسية ولكن كل واحده من هذه الاتفاقيات تعاني اما من البطيء في العمل او ان تعمل بشكل غير موثوق بها أو بشكل غير آمن.

في هذه البحث، تم تطوير اتفاقية جديدة لنقل البيانات بالاستناد على البروتوكول UPD كأتفاقية سريعة وموثوقة وأمنة، وتم تسميتها بأسم FRS-FTP . الاتفاقية المقترحة تعتمد على ثلاث مراحل، المرحلة الاولى تفرض الموثوقية بأستخدام دالة تدقيق المجاميع، المرحلة الثانية تفرض مفاهيم الامنية المتعلقة بتشفير/فك تشفير البيانات المرسلة، وحماية منفذ التراسل، والتخويل، اما المرحلة الثالثة فتتعلق بالتراسل السريع للبيانات بالاعتماد على البروتوكول UPD.

لقد تم تطوير الاتفاقية المقترحة بواسطة لغة البرمجة Visual Basic .Net والأدرة الضمنية System .Net. Sockets .